Jukka Nopanen

VIKASIETOISEN JA SKAALAUTUVAN PALOMUURIN RAKENTAMINEN AVOIMEN LÄHDEKOODIN OHJELMISTOLLA

VIKASIETOISEN JA SKAALAUTUVAN PALOMUURIN RAKENTAMINEN AVOIMEN LÄHDEKOODIN OHJELMISTOLLA

Jukka Nopanen
Opinnäytetyö
Kevät 2013
Tietotekniikan koulutusohjelma
Oulun seudun ammattikorkeakoulu

ALKULAUSE

Tämä insinöörityö on tehty syksyllä 2012 ja talvella 2013 Oulun seudun ammattikorkeakoulun Tekniikan yksikön Raahen kampuksella. Työn ohjaavana opettajana toimi Lea Hannila.

Haluan kiittää työnantajaani DNA:ta mahdollisuudesta käyttää työaikaa ja työvälineitä tämän lopputyön tekemiseen. Lisäksi haluan kiittää ohjaavana opettajana toiminutta Lea Hannilaa joustavuudesta aikataulun suhteen sekä rakentavista kommenteista.

Oulussa 29.1.2013

Jukka Nopanen

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Tietotekniikan koulutusohjelma

Tekijä: Jukka Nopanen
Opinnäytetyön nimi: Vikasietoisen ja skaalautuvan palomuurin rakentaminen avoimen lähdekoodin ohjelmistolla
Työn ohjaaja: Lea Hannila
Työn valmistumislukukausi ja -vuosi: Kevät 2013 Sivumäärä: 52

Tämän opinnäytetyön tarkoituksena oli rakentaa vikasietoinen ja skaalautuva palomuuri käyttäen avoimen lähdekoodin ohjelmistoja. Idea työn tekemiseen lähti omasta mielenkiinnosta. Vastaavaan tarkoitukseen on olemassa useita kaupallisiakin vaihtoehtoja, mutta niiden hinnat voivat muodostua kynnyskysymykseksi pienessä organisaatiossa. Opinnäytetyössä selvitettiin, tarjoaisiko avoimen lähdekoodin ohjelmisto realistisen vaihtoehdon kaupalliselle ratkaisulle.

Opinnäytetyössä rakennettiin toimiva laitteisto- ja ohjelmistotasolla kahdennettu palomuuri ja sen toimintaa testattiin eri näkökulmista. Alustana toimi kaksi HP Proliant DL380 G5 -palvelinta ja käyttöjärjestelmänä käytettiin OpenBSD:n versiota 5.2. Palomuuri rakennettiin käyttäen OpenBSD:n mukana tulevaa Packet Filter -ohjelmistoa ja kahdennus toteutettiin käyttäen CARP-protokollaa.

Työn tulosten perusteella voitiin todeta, että rakennettu ympäristö suoriutui sille tehdyistä testeistä hyvin ja se toimi koko testijakson vakaasti. Puutteita toiminnassa ei havaittu. Työn tuloksia voidaan hyödyntää myös työelämässä ja sen tekeminen auttoi syventämään omia tietoja aiheesta.

Asiasanat: tietoturva, palomuuri, avoin lähdekoodi

ABSTRACT

Oulu University of Applied Sciences
Degree programme in Information Technology

Author: Jukka Nopanen
Title of thesis: Building a scalable and redundant firewall with open source software
Supervisor: Lea Hannila
Term and year of completion: Spring 2013 Number of pages: 52

The purpose of this Bachelor's thesis work was to build a scalable and redundant firewall using open source software. The idea for this work came from own interest into the subject. There are also several different brands of commercial solutions available, but the high cost can be an obstacle for smaller organizations. This thesis work was aimed in finding out if open source software was a realistic choice.

A functional and redundant, on both hardware and software level, firewall was built in this thesis work. Functionality of the firewall was also tested using different methods. Two HP Proliant DL380 G5 servers were used as the hardware platform and OpenBSD 5.2 was used as the operating system. Firewall was built with Packet Filter that is shipped with OpenBSD and redundancy was configured using CARP protocol.

As the result of this thesis work it was found out at that the built environment managed to cope with the tests ran against it and that it worked fine through the test period. No shortcomings were found. The results can also be used in work environment. Making of this thesis work has also helped in gaining deeper knowledge of the subject.

Keywords: Information Security, Firewall, Open Source

SISÄLLYS

LYHENNELUETTELO

ARP	Address Resolution Protocol Protokolla, jolla Ethernet-verkoissa selvitetään IP-osoitetta vastaava MAC-osoite.
CARP	Common Address Redundancy Protocol Protokolla, jonka avulla useampi laite samassa aliverkossa voi jakaa saman IP-osoitteen mahdollistaen vikasietoisen palvelun.
FTP	File Transfer Protocol TCP-protokollaa käyttävä tiedostonsiirtoprotokolla kahden tietokoneen välille.
HTTP	Hyper Text Transfer Protocol Tiedonsiirtoprotokolla, jota selaimet ja WWW-palvelimet käyttävät.
HTTPS	Hypertext Transfer Protocol Secure Suojattu tiedonsiirtoprotokolla, jota selaimet ja WWW-palvelimet käyttävät.
ICMP	Internet Control Message Protocol TCP/IP-pinon kontrolliprotokolla, jolla lähetetään nopeasti viestejä koneesta toiseen.
IP	Internet Protocol TCP/IP-mallin Internet-kerroksen protokolla, joka huolehtii IP-tietoliikennepakettien toimittamisesta perille pakettikytkentäisessä Internet-verkossa.
ISO	International Organization for Standardization ISO on kansainvälinen standardisoimisjärjestö.
MAC	Media Access Control IEEE 802-verkoissa (esimerkiksi Ethernet) verkon varaamisen ja itse liikennöinnin hoitava osajärjestelmä.

NAT Network Address Translation
Osoitteenmuunnos, jossa yksityisiä IP-osoitteita piilotetaan yhden julkisen osoitteen taakse.

NTP Network Time Protocol
UDP-pohjainen protokolla täsmällisen aikatiedon välittämiseen tietokoneiden välillä.

OSI Open Systems Interconnection Reference Model
Pakettivälitteisen tietoliikenteen käsitemalli.

PF Packet Filter
OpenBSD:n mukana tuleva palomuuriohjelmisto.

SMTP Simple Mail Transfer Protocol
Sähköpostien välitysprotokolla.

SPOF Single Point of Failure
Yksittäinen piste, jonka vikaantuminen aiheuttaa koko järjestelmän vikaantumisen.

SSH Secure Shell
Etäyhteyksiin käytettävä salattu tietoliikenneprotokolla.

TCP Transmission Control Protocol
Tietoliikenneprotokolla, jolla luodaan luotettava yhteys tietokoneiden välille.

UDP User Datagram Protocol
Yhteydetön tietoliikenneprotokolla, jolla voidaan välittää tietoa tietokoneiden välillä.

VLAN Virtual LAN
Virtuaalinen lähiverkko on tekniikka, jolla fyysinen lähiverkko voidaan jakaa loogisiin osiin.

VRRP Virtual Router Redundancy Protocol

Protokolla, jonka avulla useampi laite samassa aliverkossa voi jakaa saman IP-osoitteen mahdollistaen vikasietoisen palvelun.

1 JOHDANTO

Tässä opinnäytetyössä rakennetaan vikasietoinen ja skaalautuva palomuuri käyttäen avoimen lähdekoodin ohjelmistoja. Tarjolla on useita kaupallisia vaihtoehtoja, mutta pienellä organisaatiolla hankintahinta voi muodostua esteeksi. Työssä tutkitaan myös palomuurin hallintaa ja sen toiminnan luotettavuutta.

Yleisin tapa suojautua verkossa tapahtuvia hyökkäyksiä vastaan on suodattaa organisaation tulevaa ja lähtevää verkkoliikennettä verkon palomuurilla reitittämällä kaikki liikenne kulkemaan palomuurin läpi. Palomuuri voi olla organisaation oma ja sijaita omissa tiloissa tai se voi olla operaattorin keskitetystä ympäristöstä ostettu palvelu.

Internet on nykyajan tärkein tietoverkko ja useimmat meistä käyttävät sitä päivittäin. Verkon käytön räjähdysmäisen kasvun myötä myös erilaiset tietomurrot, verkkohyökkäykset ja niiden yritykset ovat lisääntyneet. Hyökkääjien motiivit, taitotaso ja resurssit vaihtelevat suuresti. Toisaalta asialla voi olla nuori harrastaja, joka etsii verkosta mielenkiintoisia kohteita omaa uteliaisuuttaan ilman, että edes tajuaa kyseessä olevan rikollisesta toiminnasta. Toisaalta asialla saattaa olla ammattirikollisten joukko, joiden motiivina on rahan ansaitseminen. Myös teollisuusvakoilu voi toimia tietomurron motivaationa. (Zwicky, Cooper & Chapman, 2002, 11–14.)

1.1 Yleistä tietoturvasta

Tietoturvalla tarkoitetaan yleisesti tiedon luottamuksellisuutta, eheyttä ja saatavuutta (Miettinen 2002, 129–130). Palomuureilla suojataan ensisijaisesti tiedon luottamuksellisuutta ja eheyttä, mutta niiden toiminnalla voi olla haitallista vaikutusta myös tiedon saatavuuteen. Tätä vaikutusta voidaan minimoida käyttämällä kahdennettua ratkaisua.

Luottamuksellisuus

Tiedon luottamuksellisuus tarkoittaa sitä, että tieto on ainoastaan niiden henkilöiden ja tahojen käytettävissä, joilla on siihen oikeus. Teknisessä mielessä luottamuksellisuus tarkoittaa tietojen suojaamista luvatonta käyttöä vastaan. (Miettinen 2002, 129–130.)

Eheys

Tiedon eheys tarkoittaa, että tietoa ei häviä tai muodostu itsestään ja tieto pysyy alkuperäisenä eli virheettömänä koko tiedon elinkaaren sekä tietojenkäsittelyn eri vaiheiden ajan. Tiedon tulee olla siinä kunnossa, kuin sen alun perin suunniteltiin olevan, eikä sitä ole muutettu tarkoituksellisesti eri muotoon tai se ei ole muuttuneet esimerkiksi atk-järjestelmän toimintahäiriön johdosta. (Miettinen 2002, 129–130).

Saatavuus

Tiedon saatavuus tarkoittaa, että tarvittavat tiedot ovat käytettävissä silloin, kun niitä tarvitaan ja että tietoja voidaan käyttää niin pitkään kuin on tarpeen. Saatavuus pyritään takaamaan erilaisilla teknisillä järjestelyillä, jotta virhetilanteessa tietoa ei häviä ja että mahdollisista vioista aiheutuvat katkot tiedon saatavuuteen jäävät mahdollisimman lyhyiksi. (Miettinen 2002, 129–130).

1.2 OSI-malli ja TCP/IP-protokolla

OSI-malli (Open Systems Interconnection Reference Model) on ISO:n eli International Standards Organizationin 1970-luvun lopussa kehittämä pakettivälitteisen tietoliikenteen käsitemalli ja myös TCP/IP:n toiminta voidaan jakaa sen mukaisiin kerroksiin. OSI-malli jakautuu seitsemään kerrokseen, jotka on kuvattu taulukossa 1.

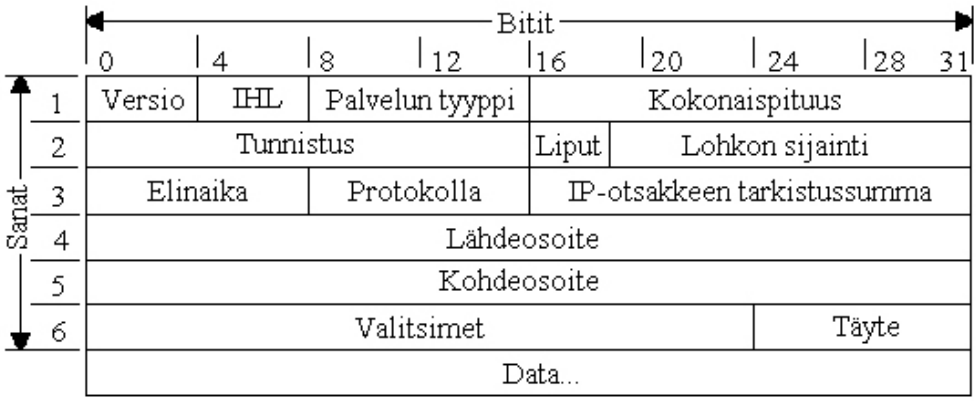
TAULUKKO 1. OSI-mallin kerrokset ja vastaavia protokollia

7. Sovelluskerros	HTTP, FTP, SMTP
6. Esitystapakerros	GIF, JPG, MPEG
5. Istuntokerros	AppleTalk, WinSock
4. Kuljetuskerros	TCP, UDP
3. Verkkokerros	IP, ICMP, IPX
2. Siirtokerros	ATM, Ethernet
1. Fyysinen kerros	Ethernet, Token Ring

OSI-mallissa jokainen kerros käyttää alemman kerroksen palveluita ja tarjoaa itse palveluita ylemmälle kerrokselle. Myös palomuurien toimintaa voidaan tarkastella OSI-mallin eri kerroksilla ja jakaa eri tyyppeihin toimintaperiaatteen mukaan.

1.2.1 IP

IP on OSI-mallin verkkokerroksen protokolla, jonka tehtävänä on huolehtia IP-pakettien toimittamisesta perille pakettikytkentäisessä verkossa. Paketit toimitetaan perille IP-osoitteiden perusteella. Kuvassa 1 on kuvattuna IP-paketin otsikko, josta osoitetiedot löytyvät. Verkossa olevat reitittimet osaavat toimittaa IP-paketin perille reititystietojen perusteella. (Stevens 1994, 33–35.)



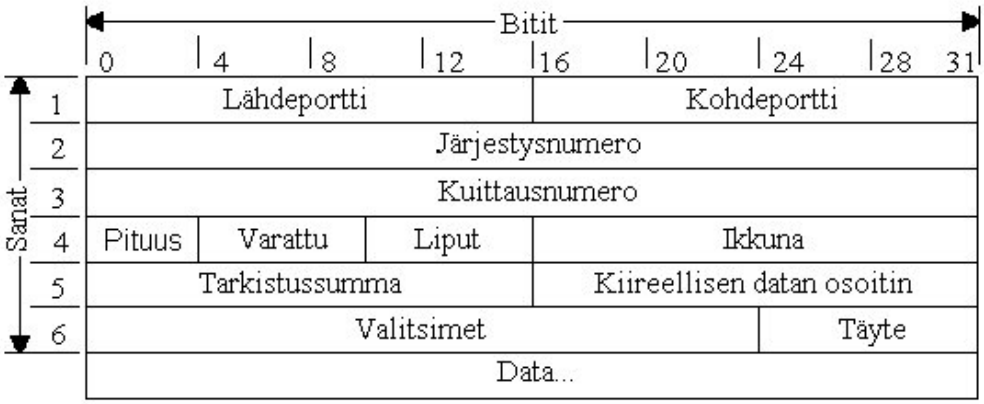
KUVA 1. IP-paketin otsikko (Stevens 1994, 34)

1.2.2 TCP

TCP (Transmission Control Protocol) on OSI-mallin kuljetuskerroksen protokolla. TCP on yhteydellinen protokolla, joka luo tietoliikenneyhteyden laitteiden välille ja huolehtii tavujen luotettavasta välittämisestä laitteiden välillä. TCP-protokolla huolehtii, että mahdolliset hävinneet paketit lähetetään uudestaan ja että paketit käsitellään oikeassa järjestyksessä. Kuvassa 2 on TCP-paketin otsikko.

TCP-yhteyden muodostaminen on kolmivaiheinen. Ensin yhteyden muodostaja lähettää SYN-paketin, seuraavaksi yhteyden toinen osapuoli vastaa siihen SYN/ACK-paketilla. Lopuksi

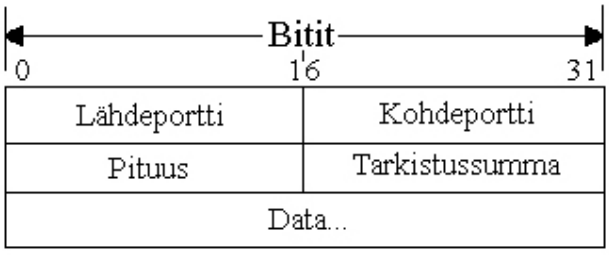
yhteyden muodostaja kuittaa sen ACK-paketilla, joka päättää kättelyvaiheen ja varsinainen datan siirto voi alkaa. (Stevens 1994, 223–227.)



KUVA 2. TCP-paketin otsikko (Stevens 1994, 225)

1.2.3 UDP

UDP (User Datagram Protocol) toimii samalla OSI-mallin kerroksella kuin TCP. UDP on yhteydetön protokolla, joten se ei tarjoa luotettavaa tiedonsiirtoa laitteiden välillä. Se ei sisällä kuittauksia eikä virheenkorjausta. Kuvassa 3 on UDP-paketin otsikko. (Stevens 1994, 143.)



KUVA 3. UDP-paketin otsikko (Stevens 1994, 144)

1.3 Palomuurien toimintaperiaatteet

Palomuureja voidaan jaotella toimintaperiaatteen mukaisesti eri tyyppeihin.

1.3.1 Tilaton pakettisuodatus

Yksinkertaisimmillaan palomuuri toimii ainoastaan tilattomana pakettisuodattimena, jolloin se toimii OSI-mallin kerroksilla 3 ja 4. Verkkoliikenteen suodatukseen käytetään ainoastaan IP-osoitetietoja (lähdeosoite ja kohdeosoite) sekä TCP- tai UDP-porttitietoja. Tilaton palomuuri vertaa jokaisen paketin otsikkotietoja palomuurisäännöstöön: jos pakettia ei ole sallittu, sitä ei välitetä eteenpäin.

Yksi tilattoman palomuurin ongelmista on dynaamiset paluupakettien portit, jolloin käytännössä yläportit (yli 1023) joudutaan avaamaan kokonaan. Tämä altistaa kaikki yläporteissa olevat palvelut verkkohyökkäyksille.

1.3.2 Tilatietoinen pakettisuodatus

Tilatietoinen palomuuri toimii samoilla OSI-mallin kerroksilla kuin tilatonkin, mutta se pitää yllä ns. tilataulua. Tilataulussa ylläpidetään listaa avatuista ja sallituista TCP- ja UDP-yhteyksistä (taulukko 2). Kun uusi yhteys avataan, tutkitaan, onko yhteys sallittu palomuurin säännöissä. Jos yhteys on hyväksytty, sen tiedot lisätään tilatauluun. Olemassa oleviin yhteyksiin kuuluvat paketit tunnistetaan lähdeosoitteen, kohdeosoitteen, lähdeportin ja kohdeportin perusteella ja päästetään läpi ilman, että koko sääntökantaa käydään läpi. Tämä nopeuttaa toimintaa ja helpottaa dynaamisten paluuporttien käsittelyä. Yhteyden purkautuessa tai vanhentuessa sen tiedot poistetaan tilataulusta ja siihen kuuluvat paketit eivät enää läpäise palomuuria.

TAULUKKO 2. Kuvitteellinen esimerkki tilataulusta

Lähdeosoite	**Lähdeportti**	**Kohdeosoite**	**Kohdeportti**
10.2.54.3	6387	192.168.90.5	22
172.18.25.5	39486	10.251.7.4	80
192.168.55.3	9386	172.31.6.3	443
10.250.4.6	16078	172.16.0.1	25

1.3.3 Sovellustason palomuuri

Sovellustason palomuuri osaa käyttää hyväkseen otsikkotietojen lisäksi paketin sisältämää dataa, joka kuuluu sovelluskerrokselle. Ne ymmärtävät yleisimpien sovellusprotokollien toimintaa (esimerkiksi FTP, HTTP, SMTP) ja osaavat tehdä päätöksiä monimutkaisemman logiikan perusteella. Säännöissä voi olla kielletty esimerkiksi yksittäiset HTTP-komennot tai jopa tietynlainen sisältö.

Sovelluspalomuuri vaatii laitteistolta parempaa suorituskykyä kuin perinteinen pakettisuodatus, mutta tarjoaa monipuolisemmat mahdollisuudet toteuttaa yrityksen tietoturvapolitiikkaa.

1.4 OpenBSD ja PF

OpenBSD on tehokas, vakaa ja turvallinen avoimen lähdekoodin BSD-pohjainen käyttöjärjestelmä. OpenBSD-projekti on keskittynyt erityisesti tietoturvaan ja se onkin säilynyt immuunina useimmille hyökkäyksille, joita sen elinaikana on ilmestynyt. OpenBSD perustuu NetBSD:hen, josta se erkani 1995 kanadalaisen Theo de Raadtin johdolla. (Palmer & Nazario, 2004, 1–12.) Versio 2.0 julkaistiin 1997 ja de Raadt jatkaa edelleen projektin johdossa. Tätä kirjoitettaessa uusin versio on 5.2, joka julkaistiin 1. marraskuuta 2012. (OpenBSD 2012.)

OpenBSD:tä käytetään tyypillisesti julkisen verkon palvelimissa, joilta edellytetään korkeaa tietoturvaa, esimerkiksi palomuureissa, nimipalvelimissa ja sähköpostipalvelimissa. Sitä ei ole rakennettu helppo ja interaktiivinen työpöytäkäyttö mielessä, joten työpöytäkäytössä OpenBSD ei ole erityisen suosittu.

1.4.1 OpenBSD ja tietoturva

OpenBSD:n tärkein tavoite on olla mahdollisimman tietoturvallinen käyttöjärjestelmä. Järjestelmän ytimen ja peruspakettien lähdekoodi on huolellisesti auditoitu ja sitä pidetään yleisesti turvallisena.

Avoimen lähdekoodin toimintamalli edistää myös tietoturvaa, koska kaikki virheet ja niiden korjaukset ovat julkisia ja kaikkien saatavilla. Lisäksi avoin lähdekoodi altistuu useamman

silmäparin tarkastelulle kuin vastaava suljetun koodin järjestelmä, joten mahdolliset piilevät ongelmat paljastuvat todennäköisemmin.

1.4.2 PF

OpenBSD:n versiosta 3.0 lähtien sen mukana on tullut palomuuriohjelmisto PF (Packet Filter). Sitä aiemmin käytettiin IPF-ohjelmistoa, mutta ohjelmistoa vaihdettiin lisensointisyistä.

PF toimii pääsääntöisesti tilatietoisena pakettisuodattimena IP-kerroksella. PF tukee myös esimerkiksi verkko-osoitekäännöstä (NAT), kuormantasausta ja liikenteen priorisointia. IP-kerroksen lisäksi se osaa toimia siltaavana pakettisuodattimena siirtokerroksella. IPv4-protokollan lisäksi myös IPv6 on tuettu. (Palmer & Nazario 2004, 285.)

1.4.3 CARP ja pfsync

CARP (Common Address Redundancy Protocol) lisättiin jakeluun versiossa 3.5 ja pfsync:n kanssa ne mahdollistavat palomuurin kahdennuksen sekä kuormantasauksen. CARP kehitettiin korvaamaan patenttiriippuvainen VRRP (Virtual Router Redundancy Protocol). CARP perustuu ryhmään palvelimia, joista yksi toimii isäntänä (master) ja muut toimivat varalla (backup). Kaikilla palvelimilla on sama IP-osoite, joka siirtyy isännältä varapalvelimelle tarvittaessa. Pfsync huolehtii tilataulun synkronisoinnista palvelinten välillä (Palmer & Nazario 2004, 119).

1.5 Multicast MAC

Normaalisti verkkokortille on tehtaalla asetettu kiinteästi täsmälähetystyypin (unicast tai singlecast) MAC-osoite, jota käytetään siirtokerroksella Ethernet-kehysten välittämiseen oikealle vastaanottajalle. Ethernet-kytkin ylläpitää MAC-osoitetaulua (taulukko 3), josta ilmenee mistä portista mikin MAC-osoite löytyy. Yhden portin takana voi olla useita MAC-osoitteita, mutta yksi MAC-osoite löytyy normaalisti vain yhden portin takaa.

TAULUKKO 3. Esimerkki MAC-osoitetaulusta

Vlan	Mac Address	Type	Ports
802	001b.7898.33b6	DYNAMIC	Po1
802	001b.7898.3442	DYNAMIC	Po2

Verkkokerroksella toimivat laitteet, kuten reitittimet ja palvelimet, ylläpitävät ARP-taulua (taulukko 4), josta selviää, mikä MAC-osoite vastaa mitäkin IP-osoitetta. Normaalisti yksi MAC-osoite voi vastata useita IP-osoitteita, mutta yhtä IP-osoitetta voi vastata vain yksi MAC-osoite.

TAULUKKO 4. Esimerkki ARP-taulusta

Device	IP Address	Phys Addr
bge0	82.128.150.1	00:08:7c:3f:df:80
bge0	82.128.150.51	00:1b:78:98:33:b6
bge0	82.128.150.52	00:1b:78:98:34:42
bge0	82.128.150.50	01:00:5e:00:01:05

Kuormaa tasaavan kahdennetun palomuurin tapauksessa kahden laitteen täytyy jakaa sama IP-osoite ja nähdä kaikki osoitteelle lähetetty liikenne. Normaalilla unicast-tyypin MAC-osoitteella tämä ei ole mahdollista, joten ratkaisuksi on kehitetty ryhmälähetys-tyyppinen (multicast) osoite. Tämä mahdollistaa saman IP-osoitteen jakamisen useamman laitteen kesken, ja kytkin osaa välittää multicast-osoitteeseen lähetetyt Ethernet-kehykset useampaan kuin yhteen porttiin.

2 MÄÄRITELMÄ

Tässä työssä rakennetaan laitteistoltaan kahdennettu palomuuri avoimen lähdekoodin ohjelmistolla ja testataan sen toimintaa ja selviytymistä vikatilanteista. Palomuurin tulisi selvitä toisen laitteen vikaantumisesta ilman merkittävää katkoa liikenteelle.

2.1 Yksinkertainen toteutus ilman vikasietoisuutta

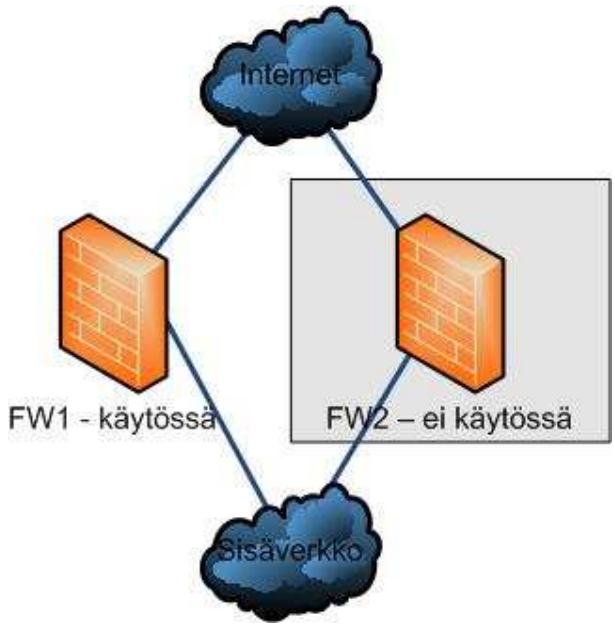
Yksinkertaisimmillaan palomuuri rakennetaan ilman kahdennusta, jolloin varsinaisia palomuurilaitteita on vain yksi kappale (kuva 4). Tällöin verkkoon muodostuu yksittäinen vikapiste (spof, single point of failure), jonka läpi kaikki liikenne kulkee. Tämän laitteen vikaantuessa organisaation kaikki liikenne katkeaa. Vian aiheuttajana voi olla esimerkiksi laitteistovika, häiriö sähkönsyötössä, ohjelmistovika tai asetusvirhe.

KUVA 4. Yksinkertainen toteutus ilman vikasietoisuutta

2.2 Kahdennettu vikasietoinen toteutus

Yleinen tapa päästä eroon yksittäisestä vikapisteestä on kahdentaa kyseinen toiminto. Palomuurin tapauksessa laitteisto voidaan kahdentaa, jolloin vikatilanteen sattuessa varalaite saadaan nopeasti käyttöön. Kahdennuksen älykkyyden mukaan palveluiden siirtyminen

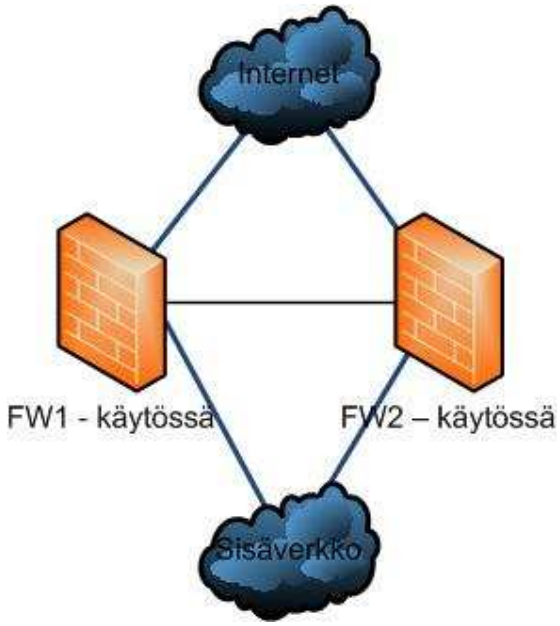
vikaantuneelta laitteelta ehjälle voi tapahtua joko automaattisesti tai manuaalisesti. Kahdentaminen lisää kuitenkin kustannuksia ja ilman kuormantasausta jättää puolet saatavilla olevista resursseista käyttämättä (kuva 5). Jos varalaite ei ole aktiivisena käytössä olevan laitteen vikaantuessa, olemassa olevat yhteydet katkeavat.



KUVA 5. Kahdennettu toteutus ilman kuormantasausta

2.3 Kahdennettu vikasietoinen ja kuormaa tasaava toteutus

Tehokkain tapa toteuttaa kahdennus ja hyödyntää sen tarjoamat mahdollisuudet on rakentaa ympäristö, jossa molemmat laitteet ovat aktiivisena ja suorittavat oman osansa kuormasta (kuva 6). Tällöin vikasietoisuuden lisäksi saavutetaan suurempi suorituskyky kuin yksittäisellä laitteella. Toisen laitteen vikaantuessa sen suorittama kuorma siirtyy ehjälle laitteelle, joka suorittaa molempien tehtävät, kunnes vikaantunut laite on korjattu. Tästä mahdollisesti aiheutuva suorituskyvyn aleneminen on huomioitava laitteiden mitoituksessa ja vikatilanteisiin varautumisessa.



KUVA 6. Kahdennettu ja kuormaa tasaava toteutus

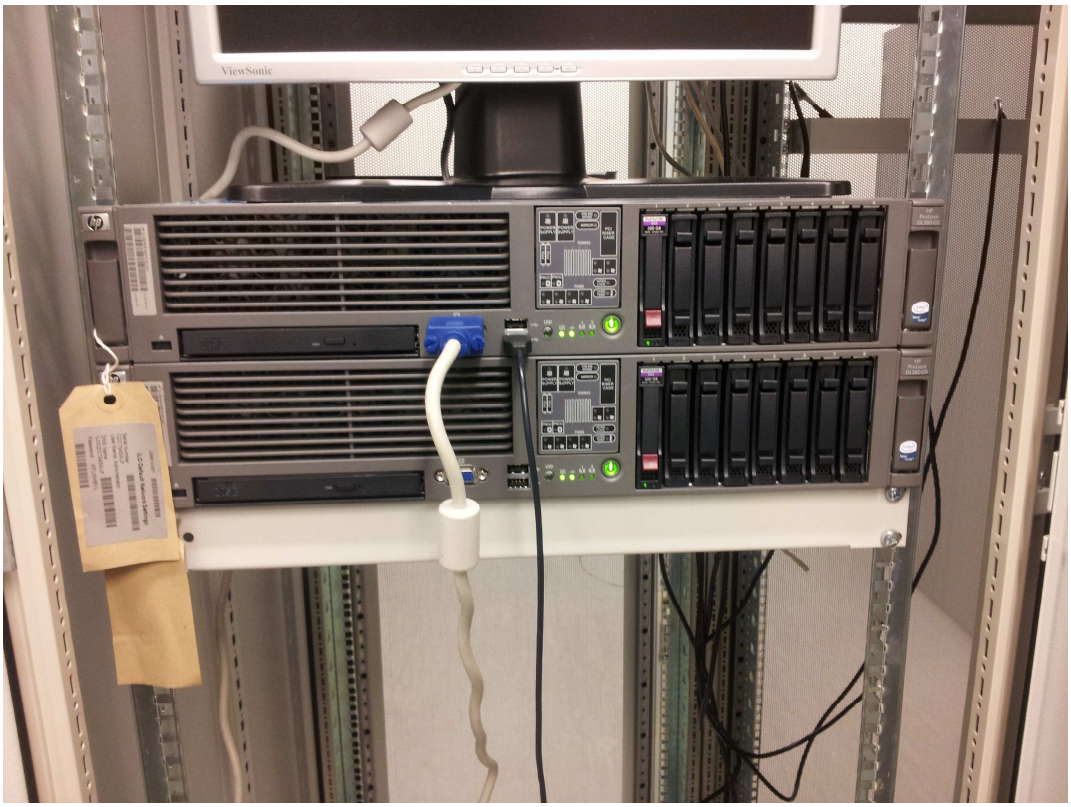
Toimiva kahdennus kuormantasauksella vaatii, että klusterin molemmat laitteet jakavat tiedon tilataulustaan. Yksittäisen TCP-yhteyden paketit saattavat kulkea välillä toisen laitteen kautta, joten sen on tunnistettava yhteys sallituksi. Tilataulujen synkronointi toteutetaan yleensä laitteiden välisellä liitännällä, jota pitkin välitetään tilataulun lisäksi tietoa laitteiden tilasta ja mahdollisesti suorituskyvystä.

2.4 Työn vaiheet

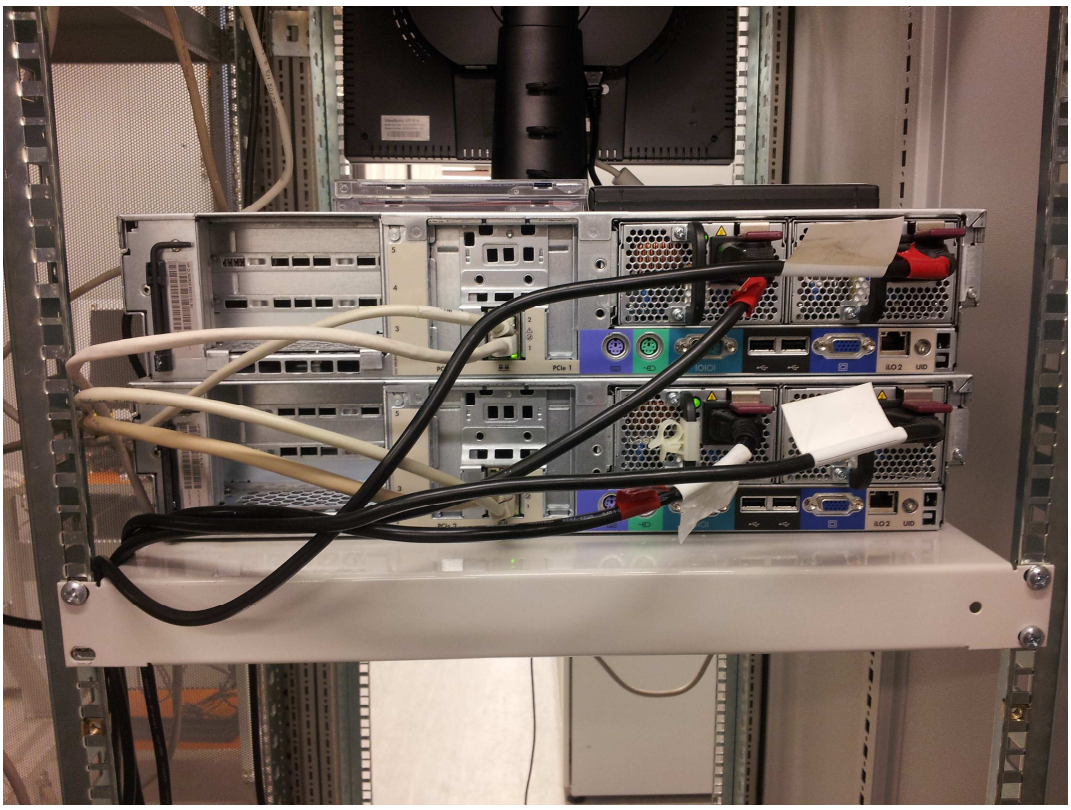
Työ jakautuu kolmeen vaiheeseen: laitteistojen fyysinen asennus, ohjelmistojen ja verkkojen konfigurointi sekä testaus.

3 TOIMINTAYMPÄRISTÖ

Tässä työssä rakennetaan vikasietoinen palomuuri käyttäen kahta HP Proliant DL380 G5 -palvelinta (kuvat 7 ja 8). Käyttöjärjestelmänä käytetään OpenBSD:n versiota 5.2. Palvelimet kytketään Ciscon Catalyst 2970 -malliseen kytkimeen. Palomuurin toimintaa testataan sijoittamalla sen taakse kaksi testipalvelinta. Palomuuria ei tulla testaamaan oikeassa tuotantoympäristössä.



KUVA 7. Palvelimet edestä kuvattuna



KUVA 8. Palvelimet takaa kuvattuna

Tarkempi kuvaus toteutuksesta löytyy luvusta 4.

4 TOTEUTUS

4.1 Laitteisto

HP Proliant DL380 G5 -palvelimissa on Intelin neliytimiset E5310-prosessorit, 4 gigatavua muistia, kaksi gigabit Ethernet -verkkoliitäntää, 300 GB:n kovalevyt ja kaksi virtalähdettä. Palvelimet asennettiin konesaliin palvelinkaappiin, sähköt kytkettiin kahdennetusti eri sulakkeiden taakse ja molemmat verkkoliitännät kytkettiin kytkimelle.

4.2 Käyttöjärjestelmä

Palvelimille asennettiin OpenBSD 5.2 -käyttöjärjestelmä, arkkitehtuuriksi valittiin 64 bittinen versio. Palvelimet nimettiin fwnode1:ksi ja fwnode2:ksi ja domainiksi asetettiin test.net (taulukko 5).

TAULUKKO 5. Palvelinten nimet ja käyttöjärjestelmäversiot

`OpenBSD fwnode1.test.net 5.2 GENERIC.MP#368 amd64`
`OpenBSD fwnode2.test.net 5.2 GENERIC.MP#368 amd64`

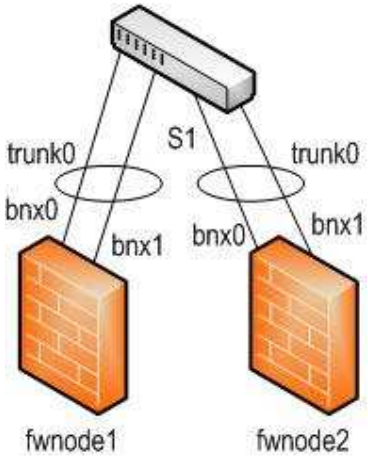
Levyt osioitiin oletusarvojen mukaisesti:

```
Filesystem Size       Used       Avail      Capacity   Mounted on
/dev/sd0a  1005M      55.5M      899M       6%         /
/dev/sd0k  239G       15.3G      212G       7%         /home
/dev/sd0d  3.9G       8.0K       3.7G       0%         /tmp
/dev/sd0f  2.0G       461M       1.4G       24%        /usr
/dev/sd0g  1005M      193M       762M       20%        /usr/X11R6
/dev/sd0h  9.8G       25.4M      9.3G       0%         /usr/local
/dev/sd0j  2.0G       2.0K       1.9G       0%         /usr/obj
/dev/sd0i  2.0G       2.0K       1.9G       0%         /usr/src
/dev/sd0e  11.8G      8.7M       11.2G      0%         /var
```

Lisäksi asennuksessa määritettiin mm. nimipalvelimet ja ajan synkronointi NTP:llä eri laitteiden logitietojen tarkastelun helpottamiseksi.

4.3 Verkkoliitännät

Palvelimet sisältävät kaksi gigabit Ethernet -verkkoliitäntää, joten ne päätettiin kahdentaa käyttäen IEEE 802.3ad -protokollaa linkkien aggregointiin. Fyysiset liitynnät bnx0 ja bnx1 yhdistettiin yhdeksi loogiseksi trunk0 liitynnäksi (kuva 9). Trunk0:lle ei määritetty IP-osoitetta.



KUVA 9. L2-tason fyysinen verkkokuva

Fwnode1 palvelimen verkkojen määrittelyt:

/etc/hostname.bnx0:

```
up description "S1 port Gi0/13"
```

/etc/hostname.bnx1:

```
up description "S1 port Gi0/14"
```

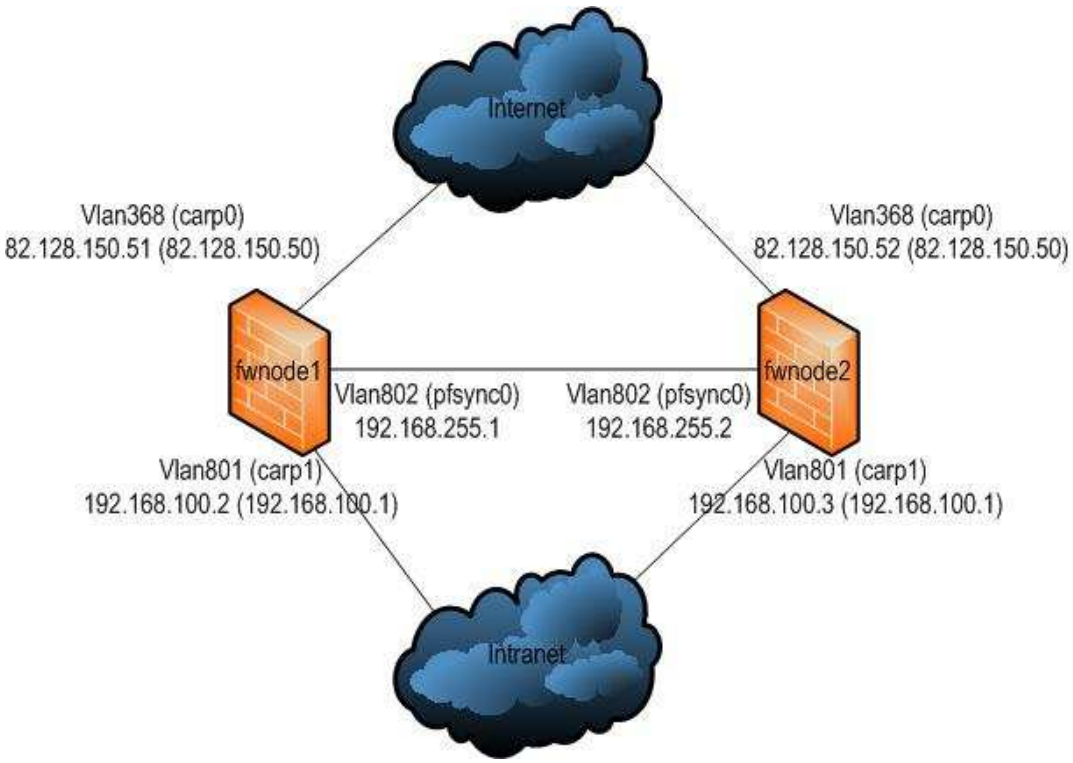
/etc/hostname.trunk0:

```
up trunkproto roundrobin trunkport bnx0 trunkport bnx1 \
description "bnx0,bnx1"
```

Todetaan muodostunut trunk0 liityntä ifconfig-komennolla:

```
trunk0:
flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST>
mtu 1500
        lladdr 00:1b:78:98:33:b6
        description: bnx0,bnx1
        priority: 0
        trunk: trunkproto roundrobin
                trunkport bnx1 active
                trunkport bnx0 master,active
        groups: trunk
        media: Ethernet autoselect
        status: active
        inet6 fe80::21b:78ff:fe98:33b6%trunk0 prefixlen 64
scopeid 0x5
```

Tarvittavat verkot luotiin loogisina virtuaaliverkkoliityntöinä (VLAN interface) trunk0 liitännän alle käyttäen IEEE 802.1Q -protokollaa virtuaaliverkkojen enkapsulointiin (kuva 10).



KUVA 10. L3-tason verkkokuva

Fwnode1 VLAN määrittelyt:

/etc/hostname.vlan368:

```
vlan 368 vlandev trunk0
inet 82.128.150.51 255.255.255.128 NONE description \
"Uplink"
```

/etc/hostname.vlan801:

```
vlan 801 vlandev trunk0
inet 192.168.100.2 255.255.255.0 NONE description \
"Intranet"
```

/etc/hostname.vlan802:

```
vlan 802 vlandev trunk0
inet 192.168.255.1 255.255.255.252 NONE description \
"Interlink"
```

Todetaan muodostunut vlan802 liityntä komennolla ifconfig:

```
vlan802: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>
mtu 1500
        lladdr 00:1b:78:98:33:b6
        description: Interlink
        priority: 0
        vlan: 802 parent interface: trunk0
        groups: vlan
        status: active
        inet6 fe80::21b:78ff:fe98:33b6%vlan802 prefixlen 64
scopeid 0x8
        inet 192.168.255.1 netmask 0xfffffffc broadcast
192.168.255.3
```

Näin toimimalla saatiin palvelinten verkkoliitännät kahdennettua palvelimen sisällä, joten yhden verkkoliitännän katoaminen ei vielä aiheuta koko palvelimen vikaantumista. Käyttämällä VLAN liityntöjä voidaan säästää verkkokorttien määrässä: vastaava ympäristö ilman virtuaaliverkkoja olisi vaatinut kuusi fyysistä verkkoliitäntää.

Koska palomuuri reitittää paketteja eri verkkojen välillä, täytyy reititys sallia molemmilla palvelimilla:

/etc/sysctl.conf

```
net.inet.ip.forwarding=1
```

4.4 Kytkimen asetukset

Palvelimet kytkettiin 28 porttiseen Cisco Catalyst 2970 -merkkiseen gigabit Ethernet -kytkimeen.

Kytkimen porttien määrittelyt:

```
interface GigabitEthernet0/13
 description fwnode1 bnx0
```

```
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 368,801,802
 switchport mode trunk
 no logging event link-status
 load-interval 60
 no snmp trap link-status
 no mdix auto
 no cdp enable
 channel-group 1 mode on
end

interface GigabitEthernet0/14
 description fwnode1 bnx1
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 368,801,802
 switchport mode trunk
 no logging event link-status
 load-interval 60
 no snmp trap link-status
 no mdix auto
 no cdp enable
 channel-group 1 mode on
end

interface GigabitEthernet0/15
 description fwnode2 bnx0
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 368,801,802
 switchport mode trunk
 no logging event link-status
 load-interval 60
 no snmp trap link-status
 no mdix auto
 no cdp enable
```

```
 channel-group 2 mode on
end

interface GigabitEthernet0/16
 description fwnode2 bnx1
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 368,801,802
 switchport mode trunk
 no logging event link-status
 load-interval 60
 no snmp trap link-status
 no mdix auto
 no cdp enable
 channel-group 2 mode on
end

interface Port-channel1
 description fwnode1 trunk
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 368,801,802
 switchport mode trunk
end

interface Port-channel2
 description fwnode2 trunk
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 368,801,802
 switchport mode trunk
end
```

4.5 Verkkoliityntöjen kahdennus palvelinten välillä

Palvelinten välinen kahdennus ja kuormantasaus toteutettiin käyttäen CARP-protokollaa. Sen avulla usea palvelin samassa paikallisverkossa voi jakaa saman IP-osoitteen. Sen ensisijainen tarkoitus on tarjota vikasietoisuutta, mutta sitä voidaan hyödyntää myös kuormantasauksessa.

Kuormaa voidaan jakaa palvelinten välillä eri algoritmien perusteella. CARP tarjoaa kaksi eri vaihtoehtoa: ARP- tai IP-perusteisen tasauksen. ARP-pohjaisessa tasauksessa CARP laskee tiivisteen lähettäjän MAC-osoitteesta ja päättää sen perusteella, mille palomuurille liikenne ohjataan. IP-pohjaisessa tasauksessa tiiviste lasketaan välitettävän IP-paketin lähde- ja kohdeosoitteista ja päätös tehdään sen perusteella. IP-pohjaisessa tasauksessa CARP-liitynnöille tulee multicast MAC -osoitteet, jotta kytkin osaa lähettää liikenteen molemmille laitteille.

Tässä ympäristössä käytetään IP-pohjaista tasausta, joten molemmat palvelimet näkevät kaiken sisään tulevan liikenteen, mutta ainoastaan toinen käsittelee sen ja välittää tarvittaessa eteenpäin. Vikasietoisuuden ansiosta toisen palvelimen hajotessa pystyy jäljelle jäänyt laite jatkamaan toimintaa häiriöittä ja kaikki liikenne ohjautuu kulkemaan sen kautta. Hallinnan helpottamiseksi kaikille verkkoliitynnöille luodaan omat fyysiset osoitteet. CARP liitynnälle luodaan looginen osoite, joka on molemmilla palvelimilla. Liitynnällä on multicast MAC -osoite, jotta kytkin osaa välittää kyseiselle IP-osoitteelle menevän liikenteen molemmille laitteille.

Koska tässä tapauksessa palvelimet toimivat palomuureina, otetaan käyttöön pre-emptiivinen tila. Tällöin jo yhden liitynnän vikaantuessa siirtyy kaikki liikenne toiselle laitteelle. Muuten saattaisi muodostua tilanne, jossa palomuurin toisen puolen verkkoliityntä olisi toiminnassa, mutta toisen puolen liityntä olisivat alhaalla. Tällöin liikennettä ei pystyttäisi välittämään verkosta toiseen.

/etc/sysctl.conf:

```
net.inet.carp.preempt=1        # 1=Enable carp(4) preemption
```

CARP-pseudolaitteiden asetukset fwnode1:llä

/etc/hostname.carp0:

```
pass secretpassword 82.128.150.50 balancing ip carpnodes \
5:100,6:0
```

Todetaan liitynnän muodostuminen:

```
carp0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>
mtu 1500
        lladdr 01:00:5e:00:01:05
        priority: 0
        carp: carpdev vlan368 advbase 1 balancing ip
                state BACKUP vhid 5 advskew 100
                state MASTER vhid 6 advskew 0
        groups: carp
        status: backup
        inet6 fe80::21b:78ff:fe98:33b6%carp0 prefixlen 64
scopeid 0x9
        inet 82.128.150.50 netmask 0xff000000 broadcast
82.255.255.255
```

/etc/hostname.carp1:

```
pass secretpassword 192.168.100.1 balancing ip carpnodes \
3:100,4:0
```

Todetaan liitynnän muodostuminen:

```
carp1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>
mtu 1500
        lladdr 01:00:5e:00:01:03
        priority: 0
        carp: carpdev vlan801 advbase 1 balancing ip
                state BACKUP vhid 3 advskew 100
                state MASTER vhid 4 advskew 0
        groups: carp
        status: backup
        inet6 fe80::21b:78ff:fe98:33b6%carp1 prefixlen 64
scopeid 0xa
```

```
        inet 192.168.100.1 netmask 0xffffff00 broadcast
192.168.100.255
```

CARP-pseudolaitteiden asetukset fwnode2:lla

/etc/hostname.carp0:

```
pass secretpassword 82.128.150.50 balancing ip carpnodes \
5:0,6:100
```

Todetaan liitynnän muodostuminen:

```
carp0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>
mtu 1500
        lladdr 01:00:5e:00:01:05
        priority: 0
        carp: carpdev vlan368 advbase 1 balancing ip
                state MASTER vhid 5 advskew 0
                state BACKUP vhid 6 advskew 100
        groups: carp
        status: master
        inet6 fe80::21b:78ff:fe98:3442%carp0 prefixlen 64
scopeid 0x9
        inet 82.128.150.50 netmask 0xff000000 broadcast
82.255.255.255
```

/etc/hostname.carp1:

```
pass secretpassword 192.168.100.1 balancing ip carpnodes \
3:0,4:100
```

Todetaan liitynnän muodostuminen:

```
carp1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>
mtu 1500
        lladdr 01:00:5e:00:01:03
        priority: 0
```

```
        carp: carpdev vlan801 advbase 1 balancing ip
                state MASTER vhid 3 advskew 0
                state BACKUP vhid 4 advskew 100
        groups: carp
        status: master
        inet6 fe80::21b:78ff:fe98:3442%carp1 prefixlen 64
scopeid 0xa
        inet 192.168.100.1 netmask 0xffffff00 broadcast
192.168.100.255
```

4.6 Tilataulun jakaminen

Tilatietoisena palomuurina PF pitää yllä tilataulua. Kahdennetussa ratkaisussa tilataulut täytyy synkronoida molempien laitteiden välillä. Ilman synkronointia laitteet eivät tietäisi toistensa läpi kulkevia yhteyksiä ja vikatilanteessa liikenteen kääntyessä laitteelta toiselle tilataulusta puuttuvat yhteydet katkeavat. Tilataulujen jakamiseen käytetään pfsync protokollaa.

Pfsync otetaan käyttöön luomalla molemmille palvelimille pfsync-pseudolaitteet:

fwnode1 /etc/hostname.pfsync0:

```
syncpeer 192.168.255.2 syncdev vlan802 up
```

fwnode2 /etc/hostname.pfsync0:

```
syncpeer 192.168.255.1 syncdev vlan802 up
```

Todetaan liityntöjen muodostuminen komennolla ifconfig:

fwnode1:

```
pfsync0: flags=41<UP,RUNNING> mtu 1500
        priority: 0
        pfsync: syncdev: vlan802 syncpeer: 192.168.255.2
maxupd: 128 defer: off
        groups: carp pfsync
```

fwnode2:

```
pfsync0: flags=41<UP,RUNNING> mtu 1500
        priority: 0
        pfsync: syncdev: vlan802 syncpeer: 192.168.255.1
maxupd: 128 defer: off
        groups: carp pfsync
```

Tilataulujen synkronointi palvelinten välillä voidaan todeta vertaamalla palvelinten tilatauluja. Tilataulun saa tulostettua komennolla pfctl -ss.

fwnode1:

```
all pfsync 192.168.255.1 -> 192.168.255.2       MULTIPLE:MULTIPLE
all carp 82.128.150.51 -> 224.0.0.18       SINGLE:NO_TRAFFIC
all carp 192.168.100.2 -> 224.0.0.18       SINGLE:NO_TRAFFIC
all carp 224.0.0.18 <- 82.128.150.52       NO_TRAFFIC:SINGLE
all carp 224.0.0.18 <- 192.168.100.3       NO_TRAFFIC:SINGLE
all carp 82.128.150.52 -> 224.0.0.18       SINGLE:NO_TRAFFIC
all carp 192.168.100.3 -> 224.0.0.18       SINGLE:NO_TRAFFIC
all carp 224.0.0.18 <- 82.128.150.51       NO_TRAFFIC:SINGLE
all carp 224.0.0.18 <- 192.168.100.2       NO_TRAFFIC:SINGLE
all       tcp       82.128.150.51:22       <-       82.128.150.101:42796
ESTABLISHED:ESTABLISHED
all      tcp      192.168.100.101:22      <-      82.128.150.101:42798
ESTABLISHED:ESTABLISHED
all      tcp      82.128.150.101:42798      ->      192.168.100.101:22
ESTABLISHED:ESTABLISHED
all      tcp      192.168.100.102:80      <-      82.128.150.101:42799
FIN_WAIT_2:FIN_WAIT_2
all      tcp      82.128.150.101:42799      ->      192.168.100.102:80
FIN_WAIT_2:FIN_WAIT_2
```

fwnode2:

```
all pfsync 192.168.255.2 -> 192.168.255.1       MULTIPLE:MULTIPLE
all carp 82.128.150.52 -> 224.0.0.18       SINGLE:NO_TRAFFIC
all carp 192.168.100.3 -> 224.0.0.18       SINGLE:NO_TRAFFIC
all carp 224.0.0.18 <- 82.128.150.51       NO_TRAFFIC:SINGLE
```

```
all carp 224.0.0.18 <- 192.168.100.2       NO_TRAFFIC:SINGLE
all carp 82.128.150.51 -> 224.0.0.18       SINGLE:NO_TRAFFIC
all carp 192.168.100.2 -> 224.0.0.18       SINGLE:NO_TRAFFIC
all carp 224.0.0.18 <- 82.128.150.52       NO_TRAFFIC:SINGLE
all carp 224.0.0.18 <- 192.168.100.3       NO_TRAFFIC:SINGLE
all       tcp       82.128.150.52:22       <-       82.128.150.101:42797
ESTABLISHED:ESTABLISHED
all       tcp       192.168.100.101:22      <-       82.128.150.101:42798
ESTABLISHED:ESTABLISHED
all       tcp       82.128.150.101:42798      ->       192.168.100.101:22
ESTABLISHED:ESTABLISHED
all       tcp       192.168.100.102:80      <-       82.128.150.101:42799
FIN_WAIT_2:FIN_WAIT_2
all       tcp       82.128.150.101:42799      ->       192.168.100.102:80
FIN_WAIT_2:FIN_WAIT_2
```

Voidaan todeta, että tilataulut ovat synkronoituneet.

4.7 Sääntökannan laatiminen PF:lle

Viimeisenä vaiheena ennen kokonaisuuden testaamista oli sääntökannan laatiminen palomuurille. Tässä ympäristössä palomuurin sisäpuolelle asennettiin kaksi palvelinta osoitteisiin 192.168.100.101 ja 192.168.100.102. Palomuurin ulkopuolella oli kaksi palvelinta osoitteissa 82.128.150.101 ja 82.128.150.111. Näistä osoitteista sallittiin hallinta- ja tuotantoyhteydet seuraavan mukaisesti:

/etc/pf.conf:

```
set skip on lo
set block-policy drop
set loginterface vlan368

# Määritellään makrot:

carpdevs = "{ vlan368, vlan801 }"
pfsyncdevs = "{ vlan802 }"
palomuurit = "{ 82.128.150.51, 82.128.150.52 }"
```

```
palvelimet = "{ 192.168.100.101, 192.168.100.102 }"
hallinta = "{ 82.128.150.101, 82.128.150.111 }"

# Määritetään antispoof käyttöön:

antispoof for vlan368
antispoof for vlan801
antispoof for vlan802

# Oletussääntö estää kaiken liikenteen:

block in log all

# Sallitaan liikenne palomuurin omista liitynnöistä:

pass out on vlan368 inet from vlan368
pass out on vlan801 inet from vlan801
pass out on vlan802 inet from vlan802


# Sallitaan SSH hallintapalvelimilta ja toiselta muurilta:

pass in on vlan368 inet proto tcp from $hallinta to self
port 22 keep state (no-sync)
pass in on vlan368 inet proto tcp from $palomuurit to self
port 22 keep state (no-sync)

# Sallitaan tuotantoliikenne testipalvelimille:

pass inet proto tcp to $palvelimet port { 22, 80, 443 }
pass inet proto { tcp, udp } from { $palvelimet,
$palomuurit } to port 53
pass inet proto udp from { $palvelimet, $palomuurit } to
port 123
```

```
# Sallitaan icmp testauksen helpottamiseksi:

pass inet proto icmp

# Sallitaan carpin ja pfsyncin vaatima liikenne:

pass on $carpdevs proto carp
pass on $pfsyncdevs proto pfsync
```

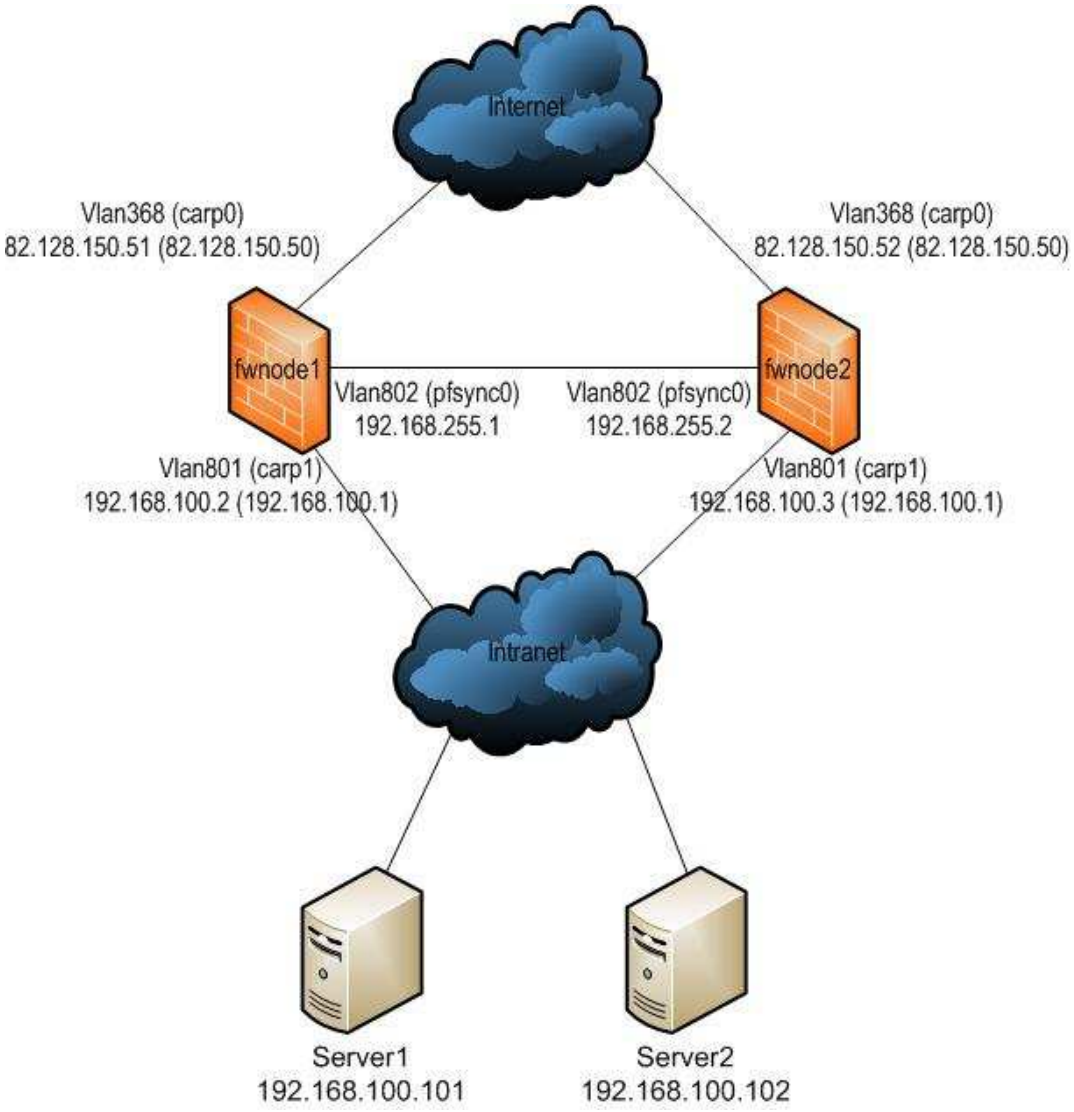
5 TESTAUS

Testaus jaettiin kolmeen osa-alueeseen: palomuurin verkkoliikenteen suodatuksen sääntöjenmukainen toiminta, kahdennuksen toiminta vikasietoisuuden kannalta sekä kahdennuksen toiminta kuormantasauksen kannalta.

Testiympäristö koostui kahdesta palomuurin taakse asennetusta palvelimesta (taulukko 6 ja kuva 11).

TAULUKKO 6. Testipalvelimet

Nimi:	**Käyttöjärjestelmä:**	**IP-osoite:**	**Malli:**
server1.test.net	Ubuntu 12.10 Linux	192.168.100.101	SunFire X4100
server2.test.net	Ubuntu 12.10 Linux	192.168.100.102	hp Proliant DL360 G4p



KUVA 11. Testiympäristö

Palvelimet oli kytketty verkkoon yhden gigabitin Ethernet-liitynnöillä. Testausta varten palvelimilla toimi SSH- ja WWW-palvelinohjelmistot. Palomuurisäännöissä oli sallittu liikenne TCP-portteihin 22 (SSH), 80 (HTTP) ja 443 (HTTPS). Lisäksi oli sallittu ICMP-protokolla testauksen helpottamiseksi.

5.1 Verkkoliikenteen suodatus

Palomuurin tietoturvallisen toiminnan testaamiseksi suoritettiin nmap verkkoskannerilla porttiskannaus molempien palvelinten kaikkiin TCP-portteihin. Samalla tarkkailtiin palomuurien

logitietoja ja kuunneltiin verkkoliikennettä testipalvelimilla käyttäen tcpdump-nimistä verkkoliikenteen tarkkailuohjelmaa.

Porttiskanneri lähettää TCP connect -tyyppisessä skannauksessa jokaiseen porttiin TCP SYN -paketin. Palomuurin ja palvelimen toiminnasta riippuen skanneri saa yksittäistä SYN-pakettia vastaavan SYN ACK -paketin, TCP RST -paketin tai ei mitään vastausta. Vasteesta riippuen skanneri osaa tulkita, oliko portti auki, kiinni vai suodatettu.

nmap TCP connect -skannaus osoitteesta 82.128.150.111 osoitteisiin 192.168.100.101 (server1) ja 192.168.100.102 (server2):

```
$ nmap -sT -p 1-65535 -T4 192.168.100.101-102

Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at
2012-12-20 11:46 EET
Interesting ports on 192.168.100.101:
Not shown: 65532 filtered ports
PORT    STATE  SERVICE
22/tcp  open   ssh
80/tcp  open   http
443/tcp closed https

Interesting ports on 192.168.100.102:
Not shown: 65532 filtered ports
PORT    STATE  SERVICE
22/tcp  open   ssh
80/tcp  open   http
443/tcp closed https

Nmap finished: 2 IP addresses (2 hosts up) scanned in
486.642 seconds
```

Skannauksen tuloksista voidaan todeta, että palomuuri toimi niin kuin pitikin. Portteihin 22, 80 ja 443 osoitettu liikenne pääsi palomuurista läpi. Portti 443 näkyy CLOSED tilassa, koska paketti pääsi palomuurista läpi mutta kyseinen portti ei ole LISTEN tilassa palvelimella. Portit 22 ja 80 vastaavat normaalisti.

Myös palomuurien logeja tarkastelemalla voidaan todeta, että muut paketit estettiin. Esimerkkinä muutama rivi:

```
Dec 20 11:54:46.186574 rule 9/(match) block in on vlan368:
82.128.150.111.28554     >     192.168.100.101.9116:     S
1422835847:1422835847(0)        win        5840        <mss
1460,sackOK,timestamp 778563755 0,nop,wscale 6> (DF)
Dec 20 11:54:46.186700 rule 9/(match) block in on vlan368:
82.128.150.111.13280     >     192.168.100.101.39575:     S
1687185884:1687185884(0)        win        5840        <mss
1460,sackOK,timestamp 778563755 0,nop,wscale 6> (DF)
Dec 20 11:54:46.186819 rule 9/(match) block in on vlan368:
82.128.150.111.12744     >     192.168.100.101.63667:     S
1381379791:1381379791(0)        win        5840        <mss
1460,sackOK,timestamp 778563755 0,nop,wscale 6> (DF)
```

5.2 Vikasietoisuus

Kahdennetussa ratkaisussa toisen palomuurin vikaantuessa kaikkien yhteyksien tulisi säilyä ja katkoksen tulisi kestää korkeintaan joitakin sekunteja. Koska pfsync huolehtii tilatiedon jakamisesta palomuurien välillä, pitäisi TCP-yhteyksien säilyä, vaikka kyseinen yhteys olisikin mennyt vikaantuneen laitteen kautta. Kun CARP huomaa toisen osapuolen katoamisen, sen pitäisi ottaa käsittelyyn myös alun perin vikaantuneen laitteen kautta kulkenut liikenne.

TCP-yhteyden säilyminen voidaan testata avaamalla SSH-yhteys palomuurin ulkopuolelta testipalvelimelle ja tarkistamalla, kumman palomuurin kautta se normaalitilanteessa menee. Tässä tapauksessa avataan yhteys osoitteesta 82.128.150.111 osoitteeseen 192.168.100.102 ja todetaan sen muodostuminen:

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address
State
tcp                  0              0  192.168.100.102:22
82.128.150.111:63630    ESTABLISHED
```

Sama yhteys näkyy molempien palomuurien tilatauluista:

```
all   tcp   82.128.150.111:63630   ->   192.168.100.102:22
ESTABLISHED:ESTABLISHED
```

Kuuntelemalla liikennettä molempien palomuurien vlan801 verkkoliitynnässä huomataan, että yhteys on valikoitunut kulkemaan fwnode1:n kautta:

```
fwnode1# tcpdump -ni vlan801 'src port 63630'
tcpdump: listening on vlan801, link-type EN10MB
12:21:44.040707 82.128.150.111.63630 > 192.168.100.102.22:
P  3640668636:3640668684(48)  ack  2359647621  win  1002
<nop,nop,timestamp 780181788 40261943> (DF) [tos 0x10]
12:21:44.041370 82.128.150.111.63630 > 192.168.100.102.22:
. ack 81 win 1002 <nop,nop,timestamp 780181788 40298188>
(DF) [tos 0x10]
```

Seuraavaksi sammutetaan fwnode1, jolloin yhteyden pitäisi siirtyä kulkemaan fwnode2:n kautta. Todetaan fwnode2:n ottaneen molemmat verkkoliitynnät haltuun syslogista:

```
carp0: state transition (vhid 6): BACKUP -> MASTER
carp1: state transition (vhid 4): BACKUP -> MASTER
```

Edelleen voidaan todeta tcpdumpilla, että liikenne kulkee oikeasti fwnode2:n vlan801 liitynnän kautta:

```
fwnode2# tcpdump -ni vlan801 'src port 63630'
tcpdump: listening on vlan801, link-type EN10MB
12:27:07.780114 82.128.150.111.63630 > 192.168.100.102.22:
P 3640687164:3640687212(48) ack 2380503141 win 1640
<nop,nop,timestamp 780505561 40351431> (DF) [tos 0x10]
12:27:07.780873 82.128.150.111.63630 > 192.168.100.102.22:
. ack 81 win 1640 <nop,nop,timestamp 780505562 40379122>
(DF) [tos 0x10]
```

Testipalvelimelta katsottuna sama TCP-yhteys on edelleen ESTABLISHED tilassa:

```
Proto Recv-Q Send-Q Local Address           Foreign Address
State
tcp                  0              0  192.168.100.102:22
82.128.150.111:63630    ESTABLISHED
```

Lisäksi voitiin todeta, että vaikka yhteys oli jatkuvassa käytössä, ei siinä huomannut minkäänlaista katkoa toisen palvelimen sammuessa.

5.3 Kuormantasaus

Palomuurien suorituskyvyn ja kuormantasauksen vaikutuksen täysimittainen testaus olisi vaatinut usean testipalvelimen sijoittamista palomuurin taakse sekä tehokkaan liikennegeneraattorin käyttöä tai esimerkiksi kuormitetun tuotantoyhteyden reitittämistä palomuurin taakse. Tässä tapauksessa tyydyttiin testaamaan yhtä aikaa muutamaa tiedoston siirtoa käyttäen HTTP protokollaa sekä ajamalla kuormittavia porttiskannauksia mahdollisimman aggressiivisella ajoituksella. Koska kaikki verkkolaitteet oli kytketty käyttäen gigabit Ethernet -liityntöjä, oli yhden yhteyden teoreettinen maksiminopeus karkeasti 100 megatavua sekunnissa.

5.3.1 Liikenteen jakautuminen laitteiden välillä

Testien perusteella voitiin todeta, että liikenne jakautui tasaisesti molempien laitteiden välille. Kuten seuraavasta tulosteesta voidaan todeta, lyhyen testijakson aikana fwnode1 välitti ulospäin 1 483 861 pakettia ja fwnode2 1 483 823 pakettia:

fwnode1:

```
# pfctl -si
Status: Enabled for 0 days 00:04:08              Debug: err

Interface Stats for vlan368            IPv4             IPv6
  Bytes In                          62281746                0
  Bytes Out                       2225286035               64
  Packets In
    Passed                          1098691                0
    Blocked                               4                0
  Packets Out
    Passed                          1483861                1
    Blocked                               0                0

State Table                          Total             Rate
  current entries                       14
  searches                           5251308         21174.6/s
  inserts                                58             0.2/s
  removals                               44             0.2/s
```

fwnode2:

```
# pfctl -si
Status: Enabled for 0 days 00:04:12              Debug: err

Interface Stats for vlan368            IPv4             IPv6
  Bytes In                          78965840                0
```

```
  Bytes Out                      2225229033                64
  Packets In
    Passed                       1413018                   0
    Blocked                            4                   0
  Packets Out
    Passed                       1483823                   1
    Blocked                            0                   0

State Table                         Total                Rate
  current entries                      14
  searches                        5875307           23314.7/s
  inserts                              58               0.2/s
  removals                             44               0.2/s
```

5.3.2 Laitteiden CPU-kuormitus

CPU-kuormitusta testattiin lataamalla molemmilta palvelimilta isoa tiedostoa HTTP:llä ja ajamalla molempia koneita kohti nmap porttiskannausta aggressiivisimmalla ajastuksella. Tätä voidaan pitää palomuurin kannalta vaikeana liikenteenä, koska jokainen paketti koskee uutta yhteyttä ja tilataulua joudutaan tarkistamaan käytännössä jokaisen paketin kohdalla. CPU:n kuormitusta tarkkailtiin vmstat komennolla. Lukemisen helpottamiseksi tulosteista poistettiin ylimääräiset kuormitusta koskevat tiedot.

fwnode1:

```
# vmstat 5
 procs    memory          traps          cpu
 r b w    avm     fre      int   sys   cs us sy id
 1 0 0  19248 3997860     3472   843   29  0  1 99
 0 0 0  19252 3997644    29208    73   16  0  9 91
 0 0 0  19252 3997380    30042    67   19  0  9 91
 0 0 0  19252 3997052    29912    75   18  0  9 91
 0 0 0  19252 3996788    28520    62   18  0 10 90
```

```
 0 0 0  19252 3996528    29830    63   17  0 10 90
 0 0 0  19252 3996256    28829    63   18  0 10 90
 0 0 0  19252 3995996    24575    67   17  0 10 90
```

fwnode2:

```
# vmstat 5
 procs    memory           traps          cpu
 r b w    avm     fre      int   sys   cs us sy id
 1 0 0  19280 3997788     3409   788   28  0  1 99
 0 0 0  19284 3996152    30249   363   66  0 11 89
 0 0 0  19284 3994324    29992   383   74  0 11 89
 0 0 0  19284 3992356    30342   415   80  0 12 88
 0 0 0  19284 3990140    27876   463   88  0 10 90
 0 0 0  19284 3987788    30722   493   94  0 12 88
```

Tuloksista nähdään, että molempien koneiden prosessorit kävivät ainoastaan pienellä kuormalla. Prosessorikuorman puolesta liikennettä olisi mennyt läpi huomattavasti enemmänkin. Lisäksi palomuurilaitteissa on nykymittapuun mukaan vanhat ja hitaat prosessorit, yksittäisen ytimen kellotaajuus on ainoastaan 1,666 GHz. Myöskään muistin määrä ei aiheuta ongelmia, 4 gigatavun muistista on käytetty ainoastaan murto-osa.

Näillä testimenetelmillä ei pystytty testaamaan palomuurin todellista läpäisykykyä tai yhtäaikaisten yhteyksien maksimimäärää. Tulosten perusteella voidaan kuitenkin todeta, että testeissä generoitu kuorma ei aiheuttanut merkittävää kuormaa ja läpäisykyky on huomattavasti suurempi.

6 JATKOKEHITYSMAHDOLLISUUDET

Tässä työssä rakennettiin toimiva perustoteutus. PF sisältää monia ominaisuuksia, joihin ei tämän työn puitteissa tutustuttu, esimerkiksi liikenteen priorisointi ja verkko-osoitemuunnos (NAT). Lisäksi palomuurisääntöjen hallintaan on olemassa apuohjelmia, joilla sääntökantaa voidaan hallita graafisen käyttöliittymän avulla. Myös vikasietoisuutta voitaisiin parantaa edelleen sijoittamalla laitteet eri palotiloihin.

6.1 PF:n lisäominaisuudet

Alla on lueteltuna PF:n tarjoamia lisäominaisuuksia.

6.1.1 NAT

Verkko-osoitteen muunnos mahdollistaa ison verkon piilottamisen yhden julkisen osoitteen taakse. IPv4-osoitteiden riittämättömyyden takia organisaation ei ole järkevää antaa julkista reititettävää IP-osoitetta jokaiselle verkossa olevalle laitteelle. Tätä varten on varattu privaatteja ns. RFC1918 osoitteita, joita ei saa reitittää julkisissa verkoissa. Kyseiset osoitteet kätketään julkisen osoitteen taakse verkko-osoitteen muunnoksessa. (Hansteen 2011, 31.)

6.1.2 Liikenteen priorisointi

ALTQ eli Alternative Queueing on joustava mekanismi, jolla voidaan rajoittaa ja priorisoida verkkoliikennettä. Normaalisti TCP/IP pino toimii fifo-periaatteella (first in, first out). Joissakin tilanteissa voi olla kuitenkin tarpeellista priorisoida yhtä liikennettä ja rajoittaa toisen tyyppistä liikennettä. (Hansteen 2011, 105–118.)

6.1.3 Logitiedot ja monitorointi

PF tarjoaa erilaisia mahdollisuuksia liikenteen logitukseen sääntökohtaisesti. Perustasolla reaaliaikaista tilannetta voi seurata tcpdumpilla pflog0 pseudolaitteesta:

```
# tcpdump -nettti pflog0
tcpdump: WARNING: snaplen raised from 116 to 160
tcpdump: listening on pflog0, link-type PFLOG
Jan 03 15:06:26.524561 rule 9/(match) block in on vlan368:
82.128.150.101.37721      >      192.168.100.102.23:      S
1801848419:1801848419(0)  win  49640  <mss  1460,nop,wscale
0,nop,nop,sackOK> (DF)
Jan 03 15:06:27.663586 rule 9/(match) block in on vlan368:
82.128.150.101.37721      >      192.168.100.102.23:      S
1801848419:1801848419(0)  win  49640  <mss  1460,nop,wscale
0,nop,nop,sackOK> (DF)
```

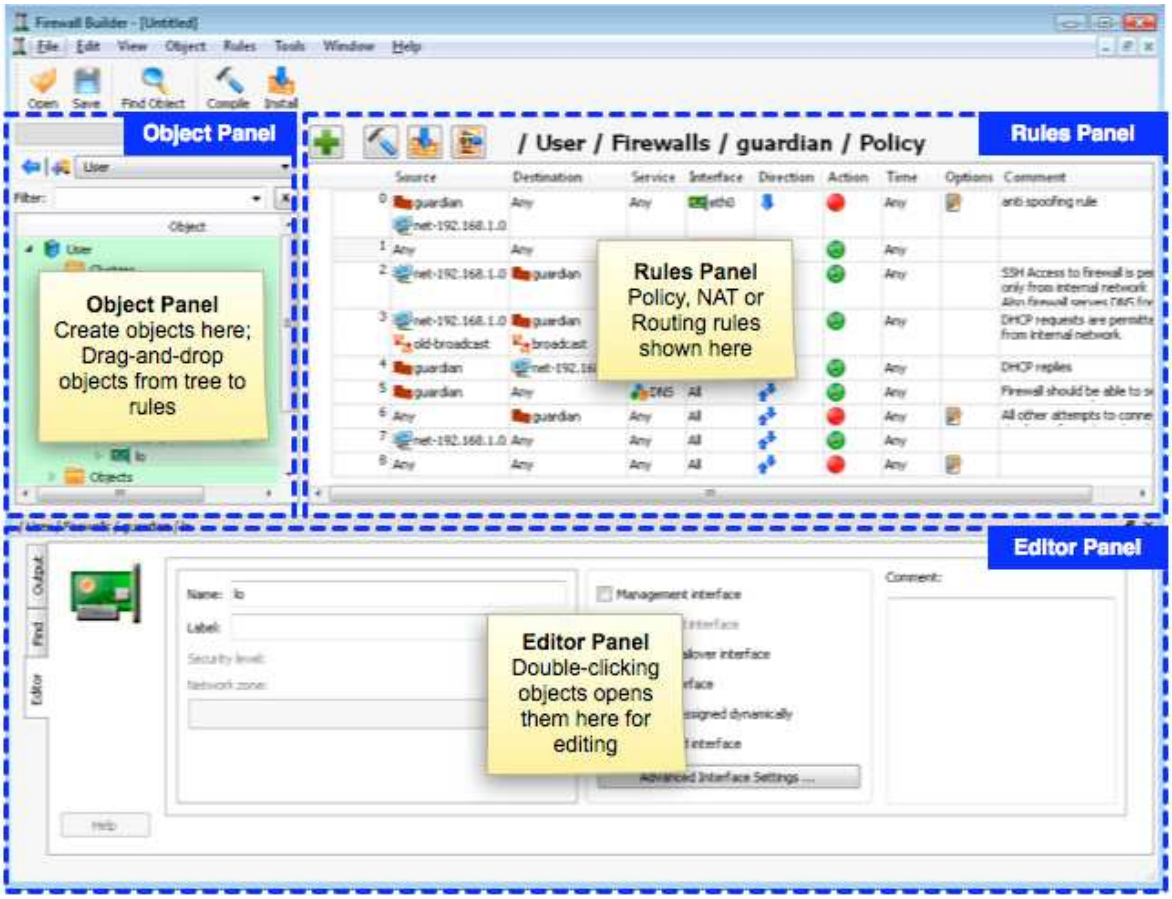
Klusteroidussa ratkaisussa paikalliset logit jakautuvat eri laitteille, jolloin logitiedot ovat hajallaan eri paikoissa. Yksi vaihtoehto olisi kerätä logit syslogilla keskitetylle logipalvelimelle, jolloin kaikkien laitteiden logit löytyisivät samasta paikasta.

Muita hyödyllisiä ohjelmia ovat mm. palomuurin reaaliaikaiseen seurantaan tarkoitettu pftop ja liikenteen graafiseen seurantaan pfstat.

6.2 Firewall Builder

Firewall Builder on avoimen lähdekoodin ohjelma, joka mahdollistaa eri palomuurien sääntökantojen luomisen ja hallinnan graafisen käyttöliittymän kautta ja se tukee myös Packet Filteriä. Perinteisesti PF:n sääntökantaa on hallittu komentoriviltä ja klusteroidussa ratkaisussa täytyy muistaa tehdä samat muutokset klusterin kaikkiin jäseniin.

Komentorivi tarjoaa kokeneelle käyttäjälle usein nopeamman tavan ja tarkemman kontrollin sääntöjen luomiseen, mutta se voi tuntua aluksi vaikealta ja työläältä.



KUVA 12. Firewall Builderin käyttöliittymä.

6.3 Muiden ohjelmien tarjoamat lisäominaisuudet

Palomuurin lisäksi samaan alustaan voidaan yhdistää useita muita toimintoja. Esimerkiksi OpenBSD:lle löytyy valmiina paketteina Snort tunkeutumisen havainnointia varten (NIDS) ja Squid www-proxy. Tietoturvan ja parhaan suorituskyvyn saamisen kannalta voidaan kuitenkin kyseenalaistaa kaikkien toimintojen yhdistäminen samoille alustoille. Verkon kriittisimpien komponenttien kohdalla parempi ratkaisu voi olla pitää eri toiminnot dedikoiduilla alustoilla.

6.4 Vikasietoisuuden parantaminen

Tässä toteutuksessa palomuurit sijaitsivat päällekkäin samassa laitekaapissa. Suurta vikasietoisuutta vaativissa kriittisissä tuotantoympäristöissä palomuurit voitaisiin sijoittaa eri palotiloihin, jopa kokonaan eri laitetiloihin. Tämä vaatii kuitenkin kaiken muunkin

verkkoinfrastruktuurin kahdentamista, riittävää tiedonsiirtokapasiteettia laitetilojen välillä ja myös palomuurilla suojattujen palveluiden kahdentamista.

7 YHTEENVETO

Työssä rakennettiin laitteistoltaan ja verkkoyhteyksiltään kahdennettu palomuuri käyttäen yleisesti saatavilla olevia Intel-pohjaisia HP:n palvelimia ja avoimen lähdekoodin OpenBSD-käyttöjärjestelmää sekä sen sisältämää Packet Filter -palomuuria.

Laitealustana käytettiin tuotannosta poistuneita palvelimia ja ne asennettiin konesaliin omaan testiympäristöön. Työssä käytetyt ohjelmat asennettiin ja konfiguroitiin alusta lähtien itse. Järjestelmän suorituskyky havaittiin hyväksi, mutta käyttämällä uusinta teknologiaa olevia palvelimia ja 10 gigabit Ethernet -liityntöjä päästäneen vielä kertaluokkaa parempaan suorituskykyyn.

Työn keskeisenä havaintona oli, että OpenBSD ja PF toimivat luotettavasti palomuurina. Kahdennus ja kuormantasaus toimivat niin kuin oli oletettavissakin ja ne tuovat selkeää lisähyötyä tuotantoympäristössä lisäämällä skaalautuvuutta ja vikasietoisuutta.

Työn tulosten perusteella voidaan todeta, että myös avoimen lähdekoodin ohjelmistolla voidaan rakentaa toimiva, suorituskykyinen ja vakaa palomuuri, joka soveltuu raskaaseenkin tuotantokäyttöön. Haittapuolena voidaan pitää hallinnan kankeutta ja virallisen tukipalvelun puutetta kaupallisiin järjestelmiin verrattuna.

Työn tekeminen oli antoisaa ja mielenkiintoista. Työn aihe lähti omasta mielenkiinnosta sitä kohtaan, mutta sen tuloksia voi hyödyntää myös työelämässä.

LÄHDELUETTELO

Hansteen, P. 2011. The Book of PF. San Francisco: No Starch Press

Miettinen, J. 2002. Yritysturvallisuuden käsikirja. Helsinki: Kauppakaari.

OpenBSD 2012. Hakupäivä 20.12.2012
http://www.openbsd.org/

Palmer, B. & Nazario, J. 2004. Secure Architectures With OpenBSD. Boston: Addison – Wesley.

Stevens, W. 1994, TCP/IP Illustrated, Volume 1: The Protocols. Boston: Addison – Wesley.

Zwicky, E., Cooper, S. & Chapman, D. 2002. Building Internet Firewalls. Sebastopol: O'Reilly.